Raport
Instytutu Automatyki i Informatyki Stosowanej
Politechniki Warszawskiej

Theoretical analysis and design concept of energy-efficient CPU – part I

(*Opracowanie podstaw teoretycznych energooszczędnego CPU – część I*)

Ewa Niewiadomska-Szynkiewicz

E-mail: ens@ia.pw.edu.pl

Raport nr: 16-3

Warszawa, maj 2016

Raport wykonany w ramach projektu badawczego NCN, numer 2015/17/B/ST6/01885

SPIS TREŚCI

1. Introduction

This report addresses the vital problem of energy efficient high performance distributed and parallel computing. The aim is to provide theoretical results that will support the ICT community with new knowledge supporting design of energy-aware resource and job management systems capable of introducing guarantees for power consumption and application performance in data centers. The main scientific challenge is to acquire new knowledge on the stochastic dynamics of data processing in HPC systems and to propose adaptive resource management algorithms efficiently exploiting new power control capabilities of hardware elements.
The attention is focused on the design and development of an energy-efficient central processing unit (CPU). The question being addressed is how to apply formal techniques to exploit dynamic voltage and frequency scaling (DVFS) and smart stand-by mechanisms, as well as high resolution hardware monitoring sensors, to optimize HPC operations. Contributions in the area of energy-efficient distributed and parallel computing will support growth of the market of environment-friendly cloud services.

2. Motivation and related works

Data centers, providing both cloud and high performance computing (HPC) services, consume increasing amounts of electrical energy. In the period from 2005 to 2010 the energy consumed by data centers worldwide rose by 56%, which was accounted to be between 1.1% and 1.5% of the total electricity use in 2010. The growth of energy consumption rises operating costs of data centers but also contributes to carbon dioxide (CO2) production. According to the analysis of current trends (gesi.org/SMARTer2020), the carbon dioxide emissions of the ICT industry are expected to exceed 2% of the global emissions, a level equivalent to the contribution of the aviation [23].

Energy efficiency (FLOPS/W) of ICT systems continues to improve (www.green500.org). However, the rate of improvement does not match the growth rate of demand for computing capacity. Unless radically new energy-aware technologies are introduced, both in hardware and software domain, it will not be possible to meet DARPA's 20-MW exaflop goal (50 GFLOPS/W) by year 2020 [2][7]. Computational power improvements are, in fact, heavily constrained by the energy budget that is necessary for driving data centers. Limiting power consumption and related thermal emission has therefore become a key problem. Based on the projections of technology development, it has been argued that the continued scaling of the available systems will eventually lead to a data center consuming more than a gigawatt of electrical power (at Exaflop level), a level that violates economic rationale for providing cloud or HPC services. Optimization of energy consumption in data centers is necessary and must be addressed in response to the market and environment protection needs [42].

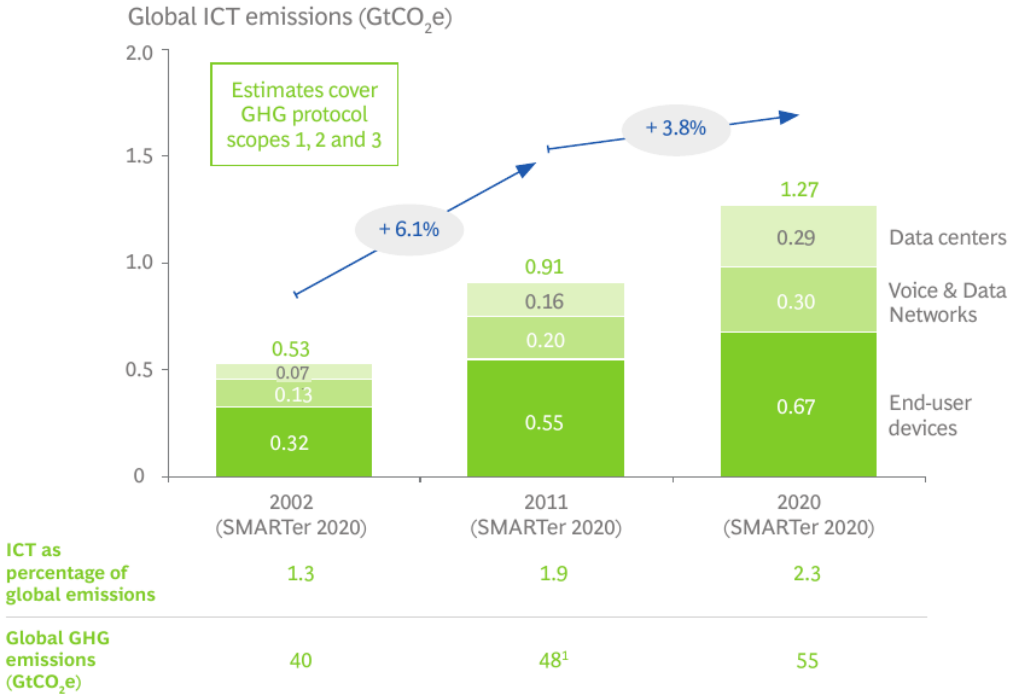

Figure 1. The role of ICT in global CO2 emissions (gesi.org/SMARTer2020).

In order to meet the challenging goals of modern cloud and high performance computing, advances in hardware layer development require immediate improvements in the design of data center control software. This includes exposing power management capabilities of hardware layer in the form of flexible and universal APIs [1]. Development of APIs and management tools is, without a doubt, essential for optimal utilization of computing resources. On the other hand, system-wide regulation of power consumption needs to be commanded by a centralized management framework, capable of collecting and processing detailed measurements, and taking real-time coordinated actions across the data center infrastructure.

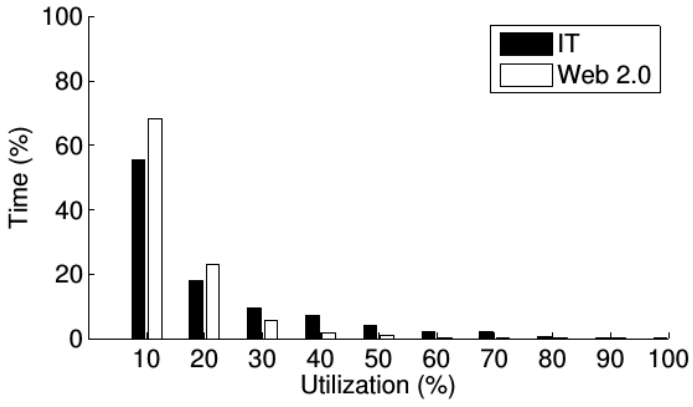

Figure 2. Server utilization histogram [3].

According to statistical data [3], the utilization of server in data centers rarely approaches 100%. Most of the time servers operate at 10-50% of their full capacity, which results from the requirements of providing sufficient quality of service provisioning margins. The over-subscription of computing resources is applied as a sound strategy to eliminate potential breakdowns caused by

traffic fluctuations or internal disruptions, e.g. hardware or software faults. A fair amount of slack capacity is also required for the purpose of maintenance tasks. Since the strategy of resource over-provisioning is, clearly, a source of energy waste, the provisioned power supply is less than the sum of the possible peak power demands of all the servers combined [9]. This, however, rises the problem of power distribution in data center. To keep the actual total power use within the available power range, servers are equipped with power budgeting mechanisms (e.g. ACPI-based) capable of limiting their power usage. The challenge of energy-efficient data center control is, therefore, to design control structure improving the utilization of computing resources and reducing energy consumption subject to quality of service constraints in highly stochastic environment (OS kernel task scheduling), capable of providing fast responses to fluctuations in application workloads. In order to reduce energy consumption the control system is required to dynamically deactivate and reactivate (by switching between low-power modes) physical computing elements (CPU/GPU, memory, interconnect) to meet the observable resource demand.

There has been a large volume of research in the area of energy-efficient control in data centers and networks. The ECONET project (www.econet-project.eu), co-developed by the authors within FP7 Programme, introduced dynamic power and performance control technologies, based on standby and performance scaling capabilities, improving energy-efficiency of wired network devices. The TREND project (www.fp7-trend.eu) aimed at integrating the activities of major European players in networking to quantitatively assess the energy demand of current and future telecom infrastructures, and to design energy-efficient, scalable and sustainable future networks. The GAMES project (www.green-datacenters.eu) aims at developing a set of innovative methodologies, metrics, Open Source ICT services and tools for the active management of energy efficiency of IT Service Centres. DEEP (www.deep-project.eu), an innovative European response to the Exascale challenge, aims at development of a novel supercomputing architecture with a matching software stack and a set of optimized grand-challenge simulation applications. DEEP-ER (www.deep-er.eu) extends the architecture of the DEEP project by a highly scalable I/O system and implements an efficient mechanism to recover application tasks that fail due to hardware errors. Furthermore, the project will leverage new memory technology to provide increased performance and power efficiency. CRESTA (www.cresta-project.eu) explores how the Exascale challenge can be met by building and exploring appropriate system software for Exascale platforms. Mont-Blanc (www.montblanc-project.eu) has a goal of designing a next-generation HPC system together with a range of embedded technologies and developing Exascale applications to be run on this new generation of HPC systems.

An extensive survey of control engineering solutions applied in various software design domains, as well as an interesting taxonomy of research directions can be found in [39]. Performance metrics for green data centers has been discussed in [48]. In [4][22] energy-aware networking techniques and the related control problems has been discussed. A survey of memory power management has been presented in [38][5]. A detailed study of performance analysis methodologies and energy monitoring mechanisms for data centers can be found in [6].
Power budgeting control systems for data centers have been proposed in [9][49][10][44][11], however the performance maximization objective has not been addressed directly. Quality of services provided by the application layer has been included in the design of power budgeting control systems in [9][46][47]. In each case the proposed control system coordinates distribution of power among virtual machines within given peak power capacity while tracking dynamic power availability and workload dynamics. To respect application performance boundaries, a combination of DVFS and CPU time allocation is used. The workload variations and power capacity changes are handled through feedback PID and MPC controllers. The problem of reducing the cooling energy consumption with DVFS and load balancing mechanisms has been addressed in [43].
In [12] a design of performance optimizing controller is proposed for a single application server. The paper describes the use of MIMO techniques to track desired CPU and memory utilizations

while capturing the related interactions between CPU and memory. System identification approach to application server controller design has been presented in [40]. Decoding rate control based on DVFS, applied for multimedia-playback system design, has been discussed in [15]. The developed feedback-control system dynamically sets the minimum possible operating frequency of CPU providing the desired throughput.
Discussion of power consumption identification problems has been presented in [25]. More precisely, it is demonstrated how inherent complexities such as multiple cores, hidden device states, and large dynamic power components can result in high prediction errors of linear models. In [13] a detailed description is given of a multi-core processor that integrates 48 cores, 4 DDR3 memory channels, and a voltage regulator controller. In particular, the paper presents hardware-level implementation and performance of power management mechanisms, allowing for independent DVFS of the cores. A DVFS technique that makes use of adaptive update intervals for optimal frequency and voltage scheduling has been proposed in [41]. The optimal control strategy is constructed to meet the workload processing deadlines given the workload arrival time.

To solve the problem of low utilization of servers in data-center running I/O-intensive applications in [26] a design of feedback controller is proposed. To adjust CPU frequency the proposed control concept relies on energy-related system-wide feedback rather than on CPU utilization levels. In [21] a technique is proposed to reduce memory bus and memory bank contention by DVFS-based control of thread execution on each core. Process model identification technique applied for the purpose of CPU frequency control design has been presented in [50]. Based on stochastic workload characterization, a feedback control design methodology is developed that leads to stochastic minimization of performance loss. The optimal design of the controller is formulated as a problem of stochastic minimization of runtime performance error for selected classed of applications. In [14] a supervised learning technique is used in DVFS to predict the performance state of the processor for each incoming task and to reduce the overhead of the state observation.

3. Energy-efficient CPU and HPC cluster – theoretical analysis and concept

3.1 Introduction

Most of currently available computing equipment does not implement any energy saving mechanisms. However, recently various efforts have been undertaken to develop prototypes of energy-aware ("green") computing and network devices that can operate in different modes, which differ in the power consumption, and implement dynamic power management techniques. Two methods are commonly used to reduce energy requirements of network devices:

- *smart standby* - the capability of automatically switching off the whole device or its component when there is no data to transmit,
- *dynamic power scaling* - the capability of decreasing the energy demands of a given network device by changing its performance.

Adaptive rate and low power idle are two common techniques of power scaling approaches. The *adaptive rate* (AR) method allows scaling the processing capacity of a given device. The *low power idle* (LPI) method puts a given device into low power state during short inactivity periods.
Most personal computers implement both AR and LPI techniques. The ACPI (Advanced Configuration and Power Interface) specification defines a number of energy-aware states attained via voltage and clock frequency scaling and idle states in which the processor is in the standby mode. The 802.3az standard defines the implementation of LPI for Ethernet interfaces.

3.2 Power consumption measurements

Energy-efficiency of computing strongly depends on capability of a hardware architecture to provide fine-grained information regarding its power consumption profile as well as its ability to adjust its performance configuration. Indeed, future architectures might not be able to operate all its components as full capability due to power delivery limitations. Consequently, software designers will need to make appropriate choices about how to allocate the available hardware-specific power budget. In order to stimulate energy-aware source code and algorithm design precise and high frequency metering of computing equipment should become a common practice. To improve the coverage of the power consumption measurements, new sensors should be implemented to measure the power consumption of CPU.

3.3 Dynamical models of HPC processes for workload forecasting

Efficiency of energy-aware governors controlling performance of a workstation (HPC node) depends on the accuracy of dynamical models of the controlled system. The task that should be done is to identify dynamical models of system processes based on the high-resolution measurements acquired from hardware counters and software profilers. To achieve this challenging goal a set of dedicated experiments should be designed and conducted to provide maximally informative input-output relations for a given process. The identification procedure should take into account multi-objective metrics, including energy consumption and processing performance. The constructed models will be subject to detailed analysis, both in time and frequency domain, and used to design of green governors, efficiently exploiting power consumption capabilities of an HPC node by dynamically adjusting its processing mode to the observed and forecasted workload. Developed workload predictors should be applied both in the workstation and whole HPC cluster management components to improve efficiency of resource allocation and job scheduling.

3.4 Green governor for CPU performance control

The objective of the NCN project is to design a dedicated energy-aware (or green) governor dynamically adjusting performance mode of CPU to the workload generated by a specified class of computing processes (scientific applications). To achieve this goal advanced control design techniques have to be applied, including methods of multi-objective optimal stochastic control and system identification. Due to innovative approach new control design methods are also expected to be developed. The main challenge is to construct feedback control efficiently exploiting hardware capabilities and high-resolution processing metrics observable at the system kernel level. Specifically, dynamic adjustment of performance will have to be made solely based on the metrics already accessible in the operating system. The proposed design concept does not allow for any modification of user algorithms (source codes)., the controller is required to respond only on the observable processing input-output signals (correlated to the energy consumption and processing performance). For this purpose the mentioned above workload forecasting mechanism will be used. The key design objective is simplicity and flexibility of the design allowing for efficient implementation and integration both with standard hardware drivers and power-aware job scheduling systems developed within the project.

3.5 Job placement techniques

From a high level point of view, job scheduling can be decomposed into two successive phases: first, the jobs are selected based upon prioritization among the group of pending jobs, then, the resources are selected upon which the job will be dispatched. In this task the goal is to extend the algorithms and architecture of the resources selection procedures within common scheduling algorithms and adapt them to power information or constraints.
Based on the accurate power consumption image of the various components of the HPC cluster, the new resources selection algorithms will favour the available resources that would consume the less power or that they will conform on particular power constraints as demanded by the application. For example based on the network power consumption of links and switches a job may consume more power if the allocation is scattered around multiple islands in the cluster rather than selecting nodes from one single island.
With the advent of techniques like GPU-direct RDMA and symmetric communication methods between Xeon-Phi's, there is no longer any involvement of the CPU and the host memory in the computation. Hence, in this context applications that need GPGPU or co-processors like Xeon Phi for offloading capabilities will be allowed to bypass CPUs and memory and allocate only the resources that they will be using. Under these power constraints the power consumption of the job will be greatly reduced. The unutilized resources could either be allocated to other jobs or put in 'deep sleep' state to consume less energy. The same strategy can also be applied for jobs where a significant fraction of the application can be completely offloaded to an accelerator or co-processor.
The new power-aware resources selection techniques should be implemented using the layouts framework. There will be parameters for global configuration done by the administrators, but also command-level parameters to allow users to provide power constraints for their jobs and have a control upon their jobs' power-aware resources selection.

3.6 Energy-efficient scheduling in HPC cluster

The goal is to extend jobs with energy profiles and allow the scheduler to take into account these profiles during scheduling. Data movement, network topology, network usage, DVFS scaling, are the type of information needed to better define/complete those profiles.

In general, we can distinguish:

*Network oriented **j**ob classification.*
A set of rules for classifying incoming jobs into two groups: the ones that require high speed network (HSN class) and the ones that do not require it (LSN class) can be created. This set will contain both application-specific rules (for known applications) as well as generic ones (e.g. analysis of mpirun parameters). The rules will be used to create analysis scripts, implemented as pre-scheduling plugins, that will tag the jobs with appropriate class flags, to be used by the scheduler.

- *Energy-efficient topology-aware scheduling.*
Energy profiles and grouping of jobs can reduce energy consumption of a HPC system. Furthermore, we can increase energy savings by addressing the question of network topology optimization. Hence, to achieve energy-efficient scheduling the HPC resource management system should be extended with a new functionality optimizing speeds of network links and supporting energy-efficient network reconfiguration utilizing energy profiles of jobs to minimize the energy consumption of the whole system (computing nodes and network equipment). The fast scheduler for energy-efficient topology-aware scheduling jobs will be implemented. It will augment the scheduling process with energy profiles, current speeds of links (given by the inspection scripts) and submitted network traffic profiles. Allocation of resources will be calculated based on the identified power-consumption models of the network infrastructure.

4. Modeling power management in common CPU

Let us deal with the efficiency of power management of state-of-the-art network devices to deeply understand and evaluate their internal dynamics and possible sources of inefficiency. To this purpose, we can start by focusing on the experimental characterization of common device platforms based on commercial off-the-shelf hardware and open-source software (i.e., common PCs/servers devices running the Linux operating system).

4.1 Reference architecture

Let us consider the reference architecture. In particular, taking into account some networking platforms, like Unix-based devices, etc., the common Linux based architecture described in the following will be considered as reference one.

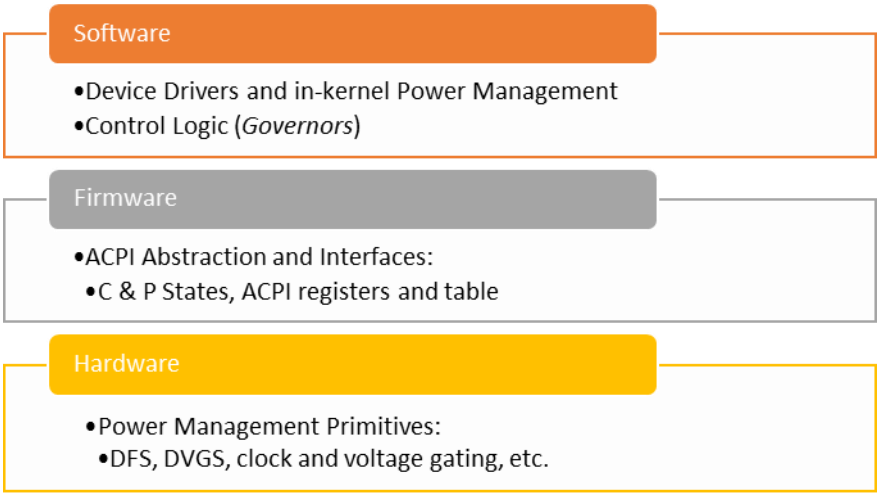

Figure 3. Architecture and main components of the power management system in off-the-shelf PCs.

At the ***hardware* level**, physical components (e.g., CPU, memories, etc.) include specific mechanisms and solutions for dynamically modulating energy consumption and performance. These mechanisms are usually based on well-known hardware techniques for power management like Dynamic Frequency Scaling (DFS), Dynamic Voltage and Frequency Scaling (DVFS), clock and supply voltage gating, etc. These mechanisms are usually implemented as power management "primitives" that can be enabled, disabled or configured by the software/firmware levels, which owns enough information to decide the most suitable trade-off between performance and energy consumption to meet application and user requirements.

The ***firmware***level (i.e., the PC BIOS – Basic Input-Output System) includes the ACPI (Advanced Configuration and Power Interface) framework, which provides a standard interface between hardware-dependent power saving techniques and software layers. Focusing on processors' power saving primitives, ACPI introduces two main types of abstraction for power management mechanisms, namely performance and idle states (*P*- and *C*-states), respectively. P- and C-states offer the possibility to the software level to neglect the specific implementation features and peculiarities of power management techniques at hardware level, and to expose them by means of intuitive functional abstractions.

At the ***software***level, and with special reference to the Linux operating system, these power

management primitives are dynamically orchestrated by specific control logics, named "*Governors*". Governors are hosted by the *cpuidle* and *cpufreq* software infrastructures that are devoted to control and manage C- and P-states, respectively.

The *cpuidle* infrastructure supports two Governors, namely *Ladder* and *Menu*, the former takes a step-wise approach by progressively setting deeper C-states depending on the time spent in the last idle period; the latter allows the CPU/Core to jump to the lowest possible idle state that does not significantly affect performance.In detail, the user[1] can request to the Governor to set the state characterized by the maximum wake-up latency.

Regarding P-states, five Governors can be applied in the *cpufreq* subsystem: *Performance*, *Powersave*, *Userspace*, *Conservative* and *Ondemand*. The *Performance* and the *Powersave* policies are static selecting the highest and the lowest frequency available, respectively. The *Userspace* Governor allows the user to set the desired working frequency. The *Conservative* Governor selects the CPU speed depending on the current utilization. In more details, every 200 ms the CPU utilization is monitored. If the utilization is above a certain threshold (called *up_threshold*) the Governor will step up the frequency to the next higher available frequency; instead, if the utilization is below the *down_threshold*, the Governor will step down the frequency to the next lower available frequency.

The *Ondemand* Governor (OG) is similar to the *Conservative* one, since it monitors the CPU utilization but uses only one threshold (σ). If the utilization is above σthen the clock frequency jumps to the maximum one, otherwise the Governor directly sets the minimum frequency available. The default value of the σ parameter is 85%, but it can be easily tuned by system administrators. The value of such threshold has clearly an important impact on CPU energy efficiency and processing performance. In fact, if σ is too small, the OG policy will set the maximum frequency even when the CPU is under-utilized, providing poor energy gain. On the contrary, a too large value of σ will make the CPU working with the minimum frequency even for medium/high workloads, considerably increasing its execution/processing times. The proposed model can support designers to choose this threshold, finding the best trade-off between energy saving and system performance. Figure 4 summarizes the CPU power management architecture in a Linux PC.

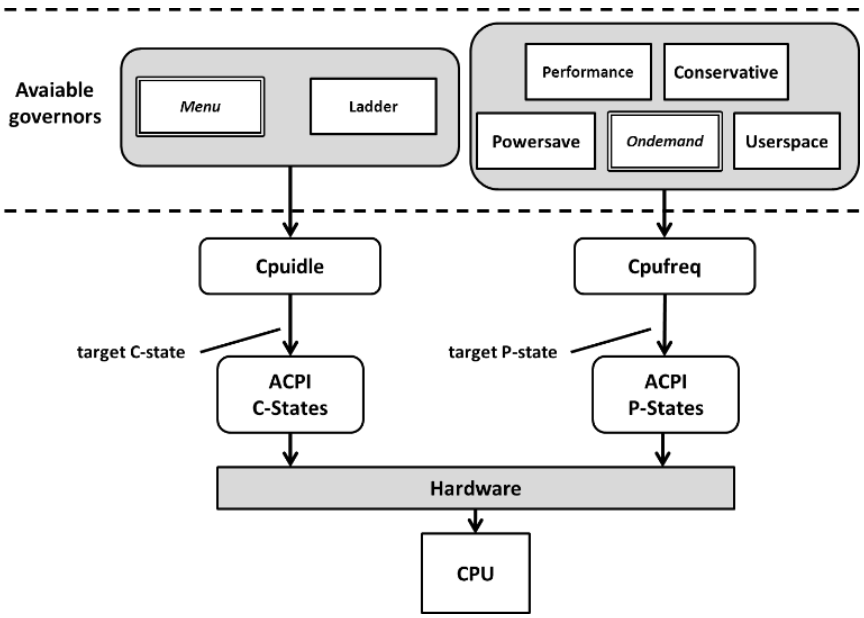

Figure 4. The CPU power management architecture in Linux OS.

[1] With the term *user*we also intend user or kernel space applications and device driver that can perform the request by behaving, in this context, as users.

4.2 Power consumption dynamics

Let us consider the system configured to work with the *Ondemand* Governor enabled to tune the clock frequency (in terms of P-states), and by leading the *Menu* Governor to throttle between the C_0 and the C_3 states. We chose to focus on this power management configuration since they are the more widely distributed in the most Linux distributions.

In order to accurately understand the power management of such devices, a number of low-level/fine-grained details should be carefully considered. In particular, it is worth noting that the workload coming from intensive network traffic stays at a time scale very close to power management dynamics, differently from the workload triggered by direct interactions of users. Thus, in order to obtain an accurate evaluation of the Linux power management system in such a scenario, we need to take into account fine-grained behavior over time, and possible inefficiencies of both hardware implementation of C- and P- states in off-the-shelf CPUs, and control logic in operating systems. In this direction, a large number of experimental measurements has been performed.

4.3 Characterizing and understanding the dynamic profiles

One of the goals of the preliminary works concerned with the design of a governor for energy-efficient CPU was to accurately represent the power consumption profile of the CPU by also taking into account transitions among C- and P-states and related inefficiencies. The main reasons for this approach are given from the not negligible energy consumption due to state transitions, especially when the CPU workload is mainly triggered by network traffic.

We decomposed the possible transitory profile of a generic ACPI-compliant CPU (working with the Ondemand and Menu Governors as introduced above) in a set of 15 descriptive states (*"SX"*), as shown in Figure 5. In Table 1 such descriptive states have been associated with the related CPU power consumption and sojourn times as obtained by the experimental results. Moreover, we considered that the system workload is driven and triggered exclusively by incoming workload.

With particular reference to the C-States, *the*$S0$ state represents the CPU while idle, (i.e. in the C_3 state). The CPU exits from $S0$ upon the reception of one or more packets or a scheduled power management event, and pass through the $S1$, $S2$ and $S3(P_i)$ states, which together represent the wake-up transitions from C_3. P_i represents the P-state/CPU frequency currently selected. We recall that the Ondemand Governor switches only between the maximum frequency and the minimum one. Thus, we indicate with $i = 0$ the cases where the P-state is P_0, and with $i = \mathrm{N}$ the ones where the P-state is P_N.

When a packet is received, it is backlogged in the network interface buffer, waiting for CPU elaboration. After exiting from the$S3(P_i)$ state, the CPU becomes active at the selected frequency P_i, and starts processing the information contained in the received packets buffered at the network interface. This state is represented by $S4(P_i)$. The system can leave the $S3(P_i)$ state only in two cases: (*i*) the backlog of software jobs requiring processing is empty (and, then, the CPU is ready to re-enter the idle mode), (*ii*) or the Ondemand Governor decides to change the selected P-state (i.e., from P_0 to P_N or vice versa).

Regarding the P-state switching, the Ondemand Governor periodically monitors the CPU utilization. By default, the monitoring process observes the CPU utilization every 200ms, and switching the operating frequency if the measured utilization exceeds or is lower than the configured threshold. Obviously, monitoring events and P-state switching can occur only during CPU activity periods. Monitoring state switching can require a CPU wake-up. If the Ondemand Governor chooses a P-state switching, the system moves through the $S5(P_i)$, $S6$ and $S7(P_j)$ states,

with the j parameter representing the new operating frequency. After this choice, the system enters the active $S4(P_j)$ state.

When the processing backlog is emptied and the CPU is ready to return to the idle mode, the system passes through the S8(P_i) and *S9* to re-enter in low power idle ($S0$). It is worth noting that $S8(P_i)$ and $S9$ are preemptible: if a new packet arrives to the system during these states, the transition towards theC_3state is aborted and the system immediately moves to the wake-up procedure (see Figure 6). Table 1 reports the estimated values of power and duration of all the system states showed in Figures 5 and 6.

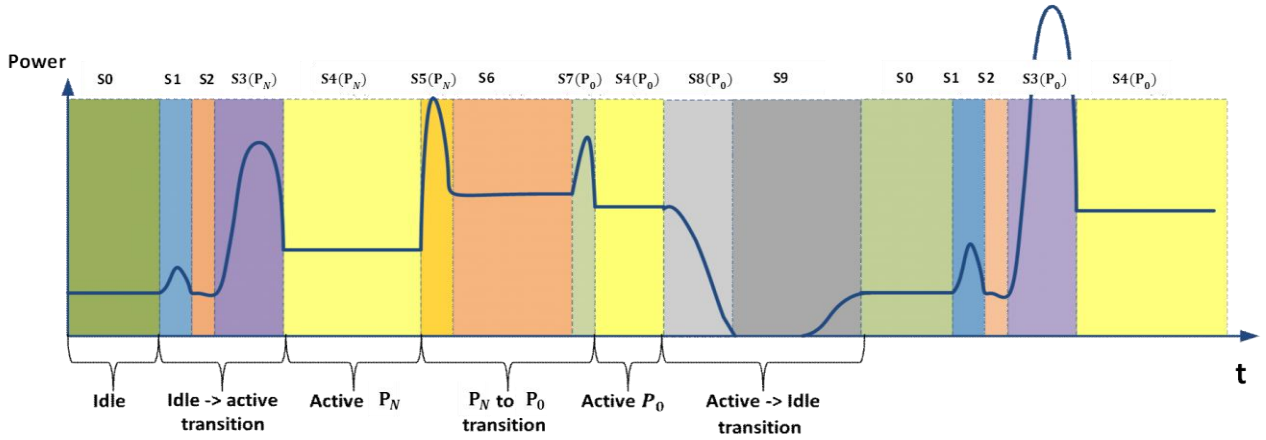

Figure 5. Example of processor power state profile.

Table 1. Estimated Values of power and duration of the system state.

State index	Power [W]	Sojourn Times [s]
S0	3	-
S1	6.7	100
S2	4.1	24
S3(0)	38	100
S3(N)	9	100
S4(0)	13.3	-
S4(N)	4.7	-
S5(0)	21	196
S5(N)	35	196
S6	18	33300
S7(0)	20	80
S7(N)	20	80
S8(0)	6	80
S8(N)	3	20
S9	1.5	140

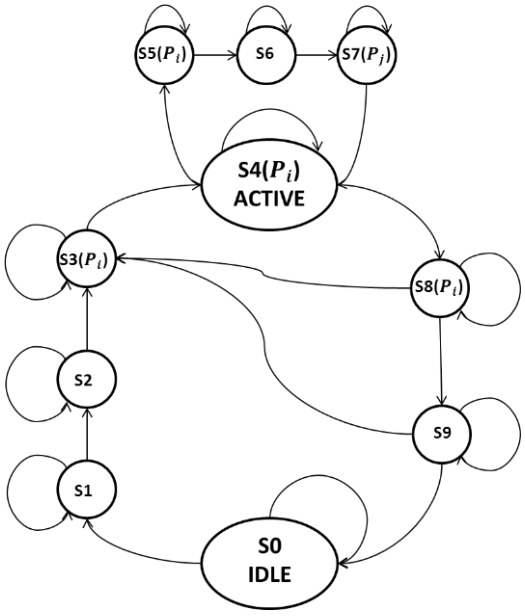

Figure 6. CPU's state diagram.

5. CPU frequency control policy – preliminary theoretical analysis

This section presents a simple model of an energy-aware service rate control system. The model is applied to design a control policy for the energy-aware processor equipped with the dynamic service rate and energy-consumption scaling mechanisms. In order to introduce energy-awareness into the control system a bi-objective performance index is used that aggregates the cost of energy consumption [W] and the service quality index [s/Mb]. Based on the applied performance model a best-response mapping is next constructed. The mapping determines service rate at which an observed workload is served at minimal cost. Analysis of the related bi-objective stochastic optimal control problem proves that the mapping defines lower bound for the energy-aware service rate control rules.

Let $U = \{u_1, \dots, u_n\}$be a set of feasible operating configurations of an energy-aware processor. Then, we consider next two functions, $Q_q: Y \times U \to R$ and $Q_p: Y \times U \to R$. The first one will be taken as a description of the service delivery cost and the second one as a description of the power-consumption cost that the processor generates under the workload $y \in Y$ while operating in the state $u \in U$.

An operating cost profile of the energy-aware processor can be described by a scalar function $Q_\gamma: Y \times U \to R$ which combines the objectives $Q_q$and Q_p. Certainly, different functions can be taken into account. Here, we apply the following one:

$$Q_\gamma(y,v) = (1-\gamma)\log(Q_q(y,v) + Q_q^0) + \gamma \log(Q_p(y,v) + Q_p^0) \tag{5.1}$$

where $\gamma \in [0,1]$ represents a weight assigned to the power-consumption cost.

Then, the following straightforward definition of performance cost function can be applied:

$$\hat{C}_\gamma(y) = \min\{Q_\gamma(z,v): y \le z, (z,v) \in Y \times U\} \tag{5.2}$$

The above function assigns a minimal value of Q_γ to the observed workload y, at which the processor is able to serve the incoming requests while operating at rate $\hat{z}_\gamma(y)$ in state $\hat{u}_\gamma(y)$ determined by means of the best-response mapping:

$$(\hat{z}_\gamma(y), \hat{u}_\gamma(y)) = \operatorname{argmin}\{Q_\gamma(z,v): y \le z, (z,v) \in Y \times U\} \tag{5.3}$$

In the remaining part of the subsection, we describe a special case where the processing configuration corresponds to a unique service rate value $v \in V$.

Thus, for a given parameter γ, the following best-response mapping is defined as follows:

$$\hat{\mu}_\gamma(y) = \operatorname{argmin}\{Q_\gamma(y,v): y \le v\} \tag{5.4}$$

The best-QoS-response and best-power-efficient-response are given by:

$$\hat{\mu}_0(y) = \operatorname{argmin}\{Q_0(y,v): y \le v\} \tag{5.5}$$

$$\hat{\mu}_1(y) = \operatorname{argmin}\{Q_1(y,v): y \le v\} \tag{5.6}$$

The proposed model of the performance profile can be applied to design a control policy for an energy-aware service rate control system. The considered system consists of a processor, its controller and a relative buffer.

The incoming service requests are placed in the buffer. After they are stored in the buffer, the processing starts at the service rate set by the processor's controller. The service rate *v*is determined by the information about the estimated workload *y* dealt by the processor since the last intervention of the controller. The service rate computation is done at the sampling instant and it is kept constant over the sampling period. These operational steps can be described by the following discrete-time dynamic system:

$$x_{k+1} = \max\{0, w_k + x_k - v_k\} \tag{5.7}$$

$$y_{k+1} = \min\{v_k, w_k + x_k\}, k = 0, \dots, N \tag{5.8}$$

where $v \in V = \{\tilde{v}_1, \dots, \tilde{v}_{np}\}, \tilde{v}_1 < \tilde{v}_2 < \cdots < \tilde{v}_{n_P}$ and *w* denotes random traffic workload generated with probability distribution *P*. This system describes the evolution of the queue length *x* and the processor's workload *y*. Given an initial buffer state x_0 the desired control policyπ, defined by a sequence of control rules $m_k(y_k) \in V$can be derived as a solution of the following control problem:

$$\begin{aligned} &\text{maximize} J_\gamma(\pi, x_0) = \lim_{N\to\infty} \mathbf{E}\left\{\sum_{k=0}^{N-1} \alpha^k Q_\gamma(y_k, v_k): w_k \sim \mathbf{P}\right\} \\ &\text{subject to} \quad x_{k+1} = \max\{0, x_k + w_k - v_k\},\ y_k = \min\{v_k, x_k + w_k\}, \\ &\text{over} \quad \pi \in \{\{m_k\}_{k=0}^{N-1} | m_k: Y \to V\}. \end{aligned} \tag{5.9}$$

Under relatively mild conditions the expected value of J_γ can be reduced if the following control input is applied:

$$\hat{\mu}_{\gamma}(y_k) \leq m_k(y_k) \leq \tilde{v}_{n_P} \tag{5.10}$$

for $k = 0, \dots, N-1$. It follows that a reasonable stationary service-rate control policy is defined by a sequence of service rate over-provisioning rules, bounded from below by function $\hat{\mu}_{\gamma}(y)$, with the level of over-provisioning specified by the value of parameter $\gamma \in [0,1]$.

REFERENCES

[1] ETSI ES 203 237 V1.1.1 (2014-03) Standard. www.etsi.org

[2] ETP4HPC Strategic Research Agenda Achieving HPC leadership in Europe. www.etp4hpc.eu

[3] Barroso, Luiz André, Hölzle, Urs: The case for energy-proportional computing, *IEEE computer* 40(12), 33–37, 2007

[4] Bolla, Raffaele, et al.: Cutting the energy bills of Internet Service Providers and telecoms through power management: An impact analysis., *Computer Networks* 56(10), 2320–2342, 2012

[5] Bostoen, Tom, Mullender, Sape, Berbers, Yolande: Power-reduction Techniques for Data-center Storage Systems, *ACM Comput. Surv.* 45(3), ACM, 33:1–33:38, 2013

[6] Benedict, Shajulin: Energy-aware performance analysis methodologies for HPC architectures—An exploratory study, *Journal of Network and Computer Applications* 35(6), 1709–1719, 2012

[7] Dongarra, Jack *et al.*: The international exascale software project roadmap, *International Journal of High Performance Computing Applications*, SAGE Publications, 2011

[8] Jing, Si-Yuan, Ali, Shahzad, She, Kun, Zhong, Yi: State-of-the-art research study for green cloud computing, *The Journal of Supercomputing* 65(1), Springer, 445–468, 2013

[9] Lim, Harold, Kansal, Aman, Liu, Jie: Power budgeting for virtualized data centers, *2011 USENIX Annual Technical Conference (USENIX ATC'11)*, 59, 2011

[10] Lefurgy, Charles, Wang, Xiaorui, Ware, Malcolm: Power capping: a prelude to power shifting, *Cluster Computing* 11(2), Springer, 183–195, 2008

[11] Fan, Xiaobo, Weber, Wolf-Dietrich, Barroso, Luiz Andre: Power provisioning for a warehouse-sized computer, *ACM SIGARCH Computer Architecture News*, volume 35, 13–23, 2007

[12] Gandhi, N, Tilbury, DM, Diao, Y, Hellerstein, J, Parekh, S: Mimo control of an apache web server: Modeling and controller design, *American Control Conference, 2002. Proceedings of the 2002*, volume 6,4922–4927, 2002

[13] Howard, Jason, Dighe, Saurabh, Vangal, Sriram R, Ruhl, Gregory, Borkar, Nitin, Jain, Shailendra, Erraguntla, Vasantha, Konow, Michael, Riepen, Michael, Gries, Matthias *et al.*: A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling, *Solid-State Circuits, IEEE Journal of* 46(1), IEEE, 173–183, 2011

[14] Jung, Hwisung, Pedram, Massoud: Supervised learning based power management for multicore processors, *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 29(9),IEEE, 1395–1408, 2010

[15] Lu, Zhijian, et al.: Control-theoretic dynamic frequency and voltage scaling for multimedia workloads, *Proceedings of the 2002 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, 156–163, 2002

[16] Karbowski A., Niewiadomska-Szynkiewicz E.: *Obliczenia równoległe i rozproszone*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2001

[17] Karbowski A., Niewiadomska-Szynkiewicz E.: *Oprogramowanie równoległe i rozproszone*, Oficyna Wydawnicza Politechniki Warszawskiej, 2009

[18] Karpowicz, Michał P.: On the design of energy-efficient service rate control mechanisms: CPU frequency control for Linux, *Digital Communications-Green ICT (TIWDC), 2013 24th Tyrrhenian International Workshop on*, 1–6, 2013

[19] Karpowicz, Michał P.: Energy efficient CPU frequency control for the Linux system. *Concurrency and Computation: Practice and Experience*. Available online, 2015

[20] Karpowicz, Michał P., Malinowski, Krzysztof: Price-based coordinability in hierarchical systems with information asymmetry, *Control and Cybernetics* 42(3), 85–110, 2013

[21] Kondo, Masaaki, Sasaki, Hiroshi, Nakamura, Hiroshi: Improving Fairness, Throughput and Energy-efficiency on a Chip Multiprocessor Through DVFS, *SIGARCH Comput. Archit. News* 35(1), ACM,
31–38, 2007
[22] Kołodziej, Joanna, Khan, Samee Ullah, Wang, Lizhe, Kisiel-Dorohinicki, Marek, Madani, Sajjad A, Niewiadomska-Szynkiewicz, Ewa, Zomaya, Albert Y, Xu, Cheng-Zhong: Security, energy, and performance-aware resource allocation mechanisms for computational grids, *Future Generation Computer Systems* 31, Elsevier, 77–92, 2014
[23] Koomey, Jonathan: Growth in data center electricity use 2005 to 2010, *A report by Analytical Press, completed at the request of The New York Times*, 2011
[24] Malinowski, Krzysztof: Optimization network flow control and price coordination with feedback: Proposal of a new distributed algorithm, *Computer Communications* 25(11-12), 1028–1036, 2002
[25] McCullough, John C, et al.: Evaluating the effectiveness of model-based power characterization, *USENIX Annual Technical Conf*, 2011
[26] Manousakis, Ioannis, Marazakis, Manolis, Bilas, Angelos: FDIO: A Feedback Driven Controller for Minimizing Energy in I/O-Intensive Applications, *Presented as part of the 5th USENIX Workshop on Hot Topics in Storage and File Systems*, 2013
[27] Marks M., Jantura J., Niewiadomska-Szynkiewicz E., Strzelczyk P., Góźdź K.: Heterogenous CPU&GPU Cluster for High Performance Computing in Cryptography, *Computer Science* 14(2), pp. 63-79, 2012.
[28] Marks M., Niewiadomska-Szynkiewicz E.*, Kołodziej J.:* High performance wireless sensor network localization system, *Inter. Journal of Ad Hoc and Ubiquitous Computing*, Vol. 17, No /32, ss. 122-133, 2014
[29] Niewiadomska-Szynkiewicz E., Sikora A.: A Software Tool for Federated Simulation of Wireless Sensor Networks and Mobile Ad Hoc Networks, LNCS, vol. 7133, pp. 303-313, 2012
[30] Niewiadomska-Szynkiewicz E., Energy Aware Communication Protocols for Wireless Sensor Networks, *Transactions on Computational Collective Intelligence*, LNCS, vol. 7776, Springer, pp. 135-149, 2013
[31] Niewiadomska-Szynkiewicz E., Sikora A., Arabas P., Kamola M., Malinowski K., Jaskóła P., Marks M. (2013), "Network-Wide Power Management in Computer Networks", Proc. *22nd International Teletraffic Congress Specialist Seminar on Energy Efficient and Green Networking* (22 ITC SSEEGN), Christchurch, NZ, 2013 (best paper award)
[32] Niewiadomska-Szynkiewicz E., Kozakiewicz A., Karbowski A.: *Obliczenia rozproszone w klastrach i gridach*, podręcznik multimedialny, wydawca Kopipol, 2007
[33] Niewiadomska-Szynkiewicz E., *Sikora A., Kołodziej J.:* Modeling Mobility in Cooperative Ad Hoc Networks, *Mobile Networks and Applications*, Vol. 18, Issue 5, pp 610-621, 2013
[34] Niewiadomska-Szynkiewicz E.: Localization in Wireless Sensor Networks: Classification and Evaluation of Techniques, *International Journal of Applied Mathematics & Computer Science*, vol. 22, No 2. ss. 281-297, 2012
[35] Niewiadomska-Szynkiewicz, Ewa, Sikora, Andrzej, Arabas, Piotr, Kołodziej, Joanna: Control system for reducing energy consumption in backbone computer network, *Concurrency and Computation: Practice and Experience* 25(12), 1738–1754, 2013
[36] Niewiadomska-Szynkiewicz, Ewa, Sikora, Andrzej, Arabas, Piotr, Kamola, Mariusz, Mincer, Marcin, Kołodziej, Joanna: Dynamic power management in energy-aware computer networks and data intensive computing systems, *Future Generation Computer Systems* 37, Elsevier, 284–296, 2014
[37] Niewiadomska-Szynkiewicz, Ewa: Computer simulation of flood operation in multireservoir systems, *Simulation* 80(2), SAGE Publications, 101, 2004

[38] Mittal, Sparsh: A survey of architectural techniques for DRAM power anagement, *International Journal of High Performance Systems Architecture* 4(2), Inderscience, 110–119, 2012

[39] Patikirikorala, Tharindu, Colman, Alan, Han, Jun, Wang, Liuping: A systematic survey on the design of self-adaptive software systems using control engineering approaches, *Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2012 ICSE Workshop on*, 33–42, 2012

[40] Padala, Pradeep, et al.: Automated control of multiple virtualized resources, *Proceedings of the 4th ACM European conference on Computer systems*, 13–26, 2009

[41] Salehi, Mostafa E, Samadi, Mehrzad, Najibi, Mehrdad, Afzali-Kusha, Ali, Pedram, Masoud, Fakhraie, Sied Mehdi: Dynamic voltage and frequency scheduling for embedded processors considering power/performance tradeoffs, *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 19(10), IEEE, 1931–1935, 2011

[42] Sikora, Andrzej, Niewiadomska-Szynkiewicz, Ewa: A federated approach to parallel and distributed simulation of complex systems, *International Journal of Applied Mathematics and Computer Science*, 17(1), Versita, 99–106, 2007

[43] Sarood, Osman, Miller, Phil, Totoni, Ehsan, Kalé, Laxmikant V: Cool Load Balancing for High Performance Computing Data Centers, *Computers, IEEE Transactions on* 61(12), IEEE, 1752–1764, 2012

[44] Stoess, Jan, Lang, Christian, Bellosa, Frank: Energy management for hypervisor-based virtual machines, *USENIX Annual Technical Conference*, 1–14, 2007

[45] Subramaniam, Balaji, Saunders, Winston, Scogland, Tom, Feng, Wu-chun: Trends in energy-efficient computing: A perspective from the Green500, *Green Computing Conference (IGCC), 2013 International*, 1–8, 2013

[46] Wang, Xiaorui, Wang, Yefu: Coordinating power control and performance management for virtualized server clusters, *Parallel and Distributed Systems, IEEE Transactions on* 22(2), 245–259, 2011

[47] Wang, Yefu, Wang, Xiaorui, Chen, Ming, Zhu, Xiaoyun: Partic: Power-aware response time control for virtualized web servers, *Parallel and Distributed Systems, IEEE Transactions on* 22(2), IEEE, 323–336, 2011

[48] Wang, Lizhe, Khan, Samee U: Review of performance metrics for green data centers: a taxonomy study, *The journal of supercomputing* 63(3), Springer, 639–656, 2013

[49] Wang, Xiaorui, Chen, Ming, Lefurgy, Charles, Keller, Tom W: SHIP: Scalable hierarchical power control for large-scale data centers, *Parallel Architectures and Compilation Techniques, 2009. PACT'09. 18th International Conference on*, 91–100, 2009

[50] Wu, Bin, Li, Peng: Load-aware stochastic feedback control for DVFS with tight performance guarantee, *VLSI and System-on-Chip (VLSI-SoC), 2012 IEEE/IFIP 20th International Conference on*, 231–236, Oct 2012

[51] Zhang, Qi, Cheng, Lu, Boutaba, Raouf: Cloud computing: state-of-the-art and research challenges, *Journal of Internet Services and Applications* 1(1), Springer-Verlag, 7–18, 2010